HEWLETT-PACKARD COMPANY
Intellectual Property Administration
P.O. Box 272400
Fort Collins, Colorado 80527-2400

PATENT APPLICATION

ATTORNEY DOCKET NO. 10012351-1

IN THE
UNITED STATES PATENT AND TRADEMARK OFFICE

Inventor(s):	Wenting Tang et al.	Confirmation No.:	5911
Application No.:	09/880,632	Examiner:	G.G. Todd
Filing Date:	June 12, 2001	Group Art Unit:	2157

Title: METHOD AND SYSTEM FOR A FRONT-END MODULAR TRANSMISSION CONTROL PROTOCOL (TCP) HANDOFF DESIGN IN A STREAMS BASED TRANSMISSION CONTROL PROTOCOL INTERNET PROTOCOL (TCP/IP) IMPLEMENTATION

Mail Stop Appeal Brief-Patents
Commissioner For Patents
PO Box 1450
Alexandria, VA 22313-1450

TRANSMITTAL OF APPEAL BRIEF

Transmitted herewith is the Appeal Brief in this application with respect to the Notice of Appeal filed on 7/2/2007.

The fee for filing this Appeal Brief was dealt with in the Appeal Brief filed 9/19/2005.

(complete (a) or (b) as applicable)

The proceedings herein are for a patent application and the provisions of 37 CFR 1.136(a) apply.

☐ (a) Applicant petitions for an extension of time under 37 CFR 1.136 (fees: 37 CFR 1.17(a)-(d)) for the total number of months checked below:

☐	1st Month $120	☐	2nd Month $450	☐	3rd Month $1020	☐	4th Month $1590

☐ The extension fee has already been filed in this application.

☒ (b) Applicant believes that no extension of time is required. However, this conditional petition is being made to provide for the possibility that applicant has inadvertently overlooked the need for a petition and fee for extension of time.

Please charge to Deposit Account 08-2025 the sum of 00.00. At any time during the pendency of this application, please charge any fees required or credit any over payment to Deposit Account 08-2025 pursuant to 37 CFR 1.25. Additionally please charge any fees to Deposit Account 08-2025 under 37 CFR 1.16 through 1.21 inclusive, and any other sections in Title 37 of the Code of Federal Regulations that may regulate fees.

Respectfully submitted,
Wenting Tang et al.

By: Jody C. Bishop
Attorney/Agent for Applicant(s)

Reg No. : 44,034
Date : July 2, 2007
Telephone : (214) 855-8007

I hereby certify that this document is being transmitted to the Patent and Trademark Office via eletronic filing.

Date of Transmission: July 2, 2007

HEWLETT-PACKARD COMPANY
Intellectual Property Administration
P.O. Box 272400
Fort Collins, Colorado 80527-2400

Docket No.: 10012351-1
(PATENT)

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Patent Application of:
Wenting Tang et al.

Application No.: 09/880,632 Confirmation No.: 5911

Filed: June 12, 2001 Art Unit: 2157

For: METHOD AND SYSTEM FOR A FRONT-END MODULAR TRANSMISSION CONTROL PROTOCOL (TCP) HANDOFF DESIGN IN A STREAMS BASED TRANSMISSION CONTROL PROTOCOL INTERNET PROTOCOL (TCP/IP) IMPLEMENTATION

Examiner: G. G. Todd

APPEAL BRIEF

MS Appeal Brief - Patents
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Dear Sir:

As required under § 41.37(a), this brief is filed within two months of the Notice of Appeal filed concurrently herewith, and is in furtherance of said Notice of Appeal.

The fees required under § 41.20(b)(2) were dealt with in the Appeal Brief filed September 19, 2005. No further fees are believed to be due for this brief.

This brief contains items under the following headings as required by 37 C.F.R. § 41.37 and M.P.E.P. § 1206:

I. Real Party In Interest
II Related Appeals and Interferences

I. REAL PARTY IN INTEREST

The real party in interest for this appeal is:

Hewlett-Packard Development Company, L.P., a Texas Limited Partnership having its principal place of business in Houston, Texas.

II. RELATED APPEALS, INTERFERENCES, AND JUDICIAL PROCEEDINGS

A notice of appeal was filed for co-pending U.S. Patent Application Serial No. 09/880,631 (hereafter "the '631 application") on April 5, 2007. The claims of the '631 application are rejected on the same grounds as the claims of the present application, i.e., rejected under 35 U.S.C. § 103(a) as being unpatentable over U.S. Patent No. 6,775,692 issued to Albert et al ("*Albert*") in view of U.S. Patent No. 5,774,660 issued to Brendel et al ("*Brendel*"). Thus, the appeal of the '631 application (and particularly the Board's interpretation of *Albert* and *Brendel*) may affect, be affected by, or have a bearing on the Board's decision in this appeal.

There are no other appeals, interferences, or judicial proceedings which will directly affect or be directly affected by or have a bearing on the Board's decision in this appeal.

III. STATUS OF CLAIMS

A. Total Number of Claims in Application

There are 29 claims pending in application.

B. Current Status of Claims

1. Claims canceled: None

2. Claims withdrawn from consideration but not canceled: None

3. Claims pending: 1-29

4. Claims allowed: None

5. Claims rejected: 1-29

C. Claims On Appeal

The claims on appeal are claims 1-29

IV. STATUS OF AMENDMENTS

A first Office Action was mailed for this application September 22, 2004. In response, Applicant filed an Amendment on December 22, 2004, which presented amendments to claims 3, 5, 17, and 22. A Final Office Action was then mailed April 20, 2005. Applicant did not file an amendment in response to the Final Office Action, but instead filed a Notice of Appeal on July 18, 2005 followed by a supporting Appeal Brief. The rejection at issue in that appeal was that all of claims 1-29 stood rejected under 35 U.S.C. § 102(e) as being anticipated by U.S. Patent No. 6,775,692 issued to Albert et al ("*Albert*").

In response to such Appeal Brief, the Examiner did not submit an Answer, but instead reopened prosecution and mailed an Office Action dated April 6, 2006 which raised a Restriction Requirement. Applicant traversed the Restriction Requirement in a response dated May 5, 2006, and the Restriction Requirement was withdrawn in an Office Action mailed July 27, 2006. However, the July 27, 2006 Office Action rejected the claims on new grounds, namely rejecting all of claims 1-29 under 35 U.S.C. § 103(a) as being unpatentable over *Albert* in view of U.S. Patent No. 5,774,660 issued to Brendel et al ("*Brendel*"). Applicant submitted a response on October 25, 2006 which presented an amendment to correct an informality in claim 12, and which pointed out that *Brendel* does not correct the deficiencies of *Albert* that were noted in the previous Appeal.

A Final Office Action was then mailed May 2, 2007 that maintained the rejection of claims 1-29 under 35 U.S.C. § 103(a) as being unpatentable over *Albert* in view of *Brendel*. Applicant did not file an amendment in response to the Final Office Action, but instead filed a Notice of Appeal, which this brief supports. Thus, the claims on appeal are those claims rejected in the Final Office Action mailed May 2, 2007, and a listing of those claims are provided in Appendix A hereto.

V. SUMMARY OF CLAIMED SUBJECT MATTER

The following provides a concise explanation of the subject matter defined in the separately argued claims involved in the appeal, referring to the specification by page and line number and to the drawings by reference characters, as required by 37 C.F.R. § 41.37(c)(1)(v). Each element of the claims is identified by a corresponding reference to the specification and drawings where applicable. Note that the citation to passages in the specification and drawings for each claim element does not imply that the limitations from the specification and drawings should be read into the corresponding claim element.

According to one claimed embodiment of the present invention, such as that of independent claim 1, a method of TCP state migration in a communication network comprises establishing a communication session between a client (client 210 of Figure 2) and a front-end node (front-end node 252 of Figure 2) at a first bottom TCP (BTCP) module (BTCP module 350 of Figures 3C and 5, and *see* page 7, lines 16-29 and page 24, lines 13-25 of the specification), located below a first TCP module (TCP module 320 of Figures 3B, 3C and 5, and *see* page 7, lines 16-24 and page 19, line 28 through page 20, line 27 of the specification) in a first operating system at the front-end node. The front-end node accesses a plurality of back-end web servers (back-end web servers 255, 257, and 259 of Figure 2, and *see* page 7, lines 5-15 of the specification) forming a web server cluster that contains content. The method further comprises receiving a HTTP request from the client at the first BTCP module (block 760 of Figure 7, and *see* page 8, lines 1-3 and page 28, lines 24-28 of the specification), and parsing the HTTP request to determine which back-end web server ("a selected back-end web server") in the plurality of back-end web servers can process the HTTP request (block 620 of Figure 6, and *see* page 8, lines 1-10, page 26, lines 4-10, and page 28, lines 26-28 of the specification). The selected back-end web server is not the front-end node. The method further comprises extending the communication session to the selected back-end web server by handing-off an initial TCP state of the first BTCP module to the selected back-end web server (block 630 of Figure 6, and *see* page 8, lines 11-15, page 8, line 23 through page 9, line 2, and page 26, lines 12-23 of the specification), and sending the HTTP request to the selected back-end web server. The method further comprises switching a bottom IP (BIP) module at the front-end node to a forwarding

mode, wherein packets received at the BIP module from the client are forwarded to the selected back-end web server (block 640 of Figure 6, and *see* page 8, lines 17-21 and page 26, lines 25-29 of the specification). The BIP module (BIP module 360 of Figures 3C and 5, and *see* page 24, lines 13-25 of the specification) is located below an IP module (IP module 330 of Figures 3B, 3C and 5, and *see* page 19, line 28 through page 20, line 27 of the specification) at the front-end node. The method further comprises terminating the communication session at the front-end node after the HTTP request is fully processed (*see* page 27, lines 1-5 and page 33, lines 8-16 of the specification).

In one embodiment, such as that of dependent claim 3, the back-end web server includes a second BTCP module (BTCP module 520 of Figure 5) that is located below a second TCP module (TCP module 530 of Figure 5) in a second operating system.

In one embodiment, such as that of dependent claim 5, extending the communication session to the selected back-end web server by handing-off an initial TCP state further comprises sending a SYN packet to said selected back-end web server (block 810 of Figure 8). The SYN packet is intercepted by a second BTCP module (BTCP module 520 of Figure 5, and block 830 of Figure 8). The SYN packet is originally sent from the client to the front-end node in requesting the communication session. The SYN packet is stored at the first BTCP module. The extending further comprises including an initial sequence number within the SYN packet that enables the second BTCP module to understand a proper TCP state of the first BTCP module in the communication session (block 820 of Figure 8). The extending further comprises receiving a SYN/ACK packet from the selected back-end web server (block 850 of Figure 8), the SYN/ACK packet updated by the second BTCP module to reflect the proper TCP state of the first BTCP module (block 860 of Figure 8). And, the extending further comprises sending an ACK packet from the first BTCP module to the selected back-end web server (block 880 of Figure 8), the ACK packet originally sent from the client to the front-end node in establishing the communication session.

In one embodiment, such as that of dependent claim 6, the method further comprises sending response packets from the selected back-end web server to the client in a communication

path that does not include the front-end node by changing headers of the response packets such that it appears that the source of said response packets is the first BTCP in its proper TCP state (*see* Figure 2, and *see* page 17, lines 5-15 of the specification).

In one embodiment, such as that of dependent claim 7, terminating the communication session further comprises intercepting TCP control packets from a second TCP module (TCP module 530 of Figure 5) located at the selected back-end web server at the second BTCP module (*see* page 33, lines 8-11 of the specification); sending the TCP control packets to the first BTCP module from the second BTCP module (page 33, lines 8-11 of the specification); sending the TCP control packets to the client from the first BTCP module (page 33, lines 11-14 of the specification); and terminating the communication session at the front-end node and the back-end web server (*see* page 9, lines 4-15 of the specification).

According to another claimed embodiment, such as that of independent claim 11, a method of TCP state migration in a communication network comprises receiving a request from a client (client 210 of Figure 2) for establishing a communication session at a first bottom TCP (BTCP) module (BTCP module 350 of Figures 3C and 5, and *see* page 7, lines 16-29 and page 24, lines 13-25 of the specification) located at a front-end node (front-end node 252 of Figure 2). The front-end node accesses a plurality of back-end web servers containing content (back-end web servers 255, 257, and 259 of Figure 2, and *see* page 7, lines 5-15 of the specification), wherein the content is partially replicated between each of the plurality of back-end web servers. The communication session is established for the transfer of data contained within the content to the client. The method further comprises establishing the communication session between the client and the first BTCP module. The first BTCP module is located below a first TCP module (TCP module 320 of Figures 3B, 3C, and 5, and *see* page 7, lines 16-24 and page 19, line 28 through page 20, lines 27 of the specification) in a first operating system at the front-end node. The method further comprises receiving a HTTP request from the client at the first BTCP module (block 760 of Figure 7, and *see* page 8, lines 1-3 and page 28, lines 24-28 of the specification), and parsing the HTTP request to determine which back-end web server ("a selected back-end web server") in the plurality of back-end web servers contains the data in order to process the HTTP request (block 620 of Figure 6, and *see* page 8, lines 1-10, page 26,

lines 4-10, and page 28, lines 26-28 of the specification). The selected back-end web server is not the front-end node. The method further comprises extending the communication session to the selected back-end web server by handing-off an initial TCP state of the first BTCP module to a second BTCP module located at the selected back-end web server (block 630 of Figure 6, and *see* page 8, lines 11-15, page 8, lines 23 through page 9, line 2, and page 26, lines 12-23 of the specification). The initial TCP state is associated with the communication session between the client and the first BTCP module, and the second BTCP module (BTCP module 520 of Figure 5) is located below a second TCP module (TCP module 530 of Figure 5) in a second operating system at the selected back-end web server. The method further comprises sending the HTTP request to the selected back-end web server. The method further comprises switching a bottom IP (BIP) module in the front-end node to a forwarding mode, wherein packets, from the client, received at the front-end node are intercepted by the BIP module and forwarded to the selected back-end web server (block 640 of Figure 6, and *see* page 8, lines 17-21 and page 26, lines 25-29 of the specification). The BIP module (BIP module 360 of Figures 3C and 5, and *see* page 24, lines 13-25 of the specification) is located below an IP module (IP module 330 of Figures 3C, 3C, and 5, and *see* page 19, line 28 through page 20, line 27 of the specification) in the front-end node. The BIP module changes the destination IP addresses of the packets to the selected back-end web server (page 32, line 10 through page 33, line 6 of the specification). The method further comprises terminating the communication session after the HTTP request has been fully processed (page 26, lines 1-5 and page 33, lines 8-16 of the specification).

In one embodiment, such as that of dependent claim 12, extending the communication session further comprises storing a SYN packet sent from the client to the front-end node, the SYN packet requesting the communication session (page 9, lines 16-19 of the specification); storing an ACK packet sent from the client to the front end node in establishing the communication session (page 9, lines 19-22 of the specification); sending the SYN packet to the selected back-end web server so that it appears that the SYN packet originated from the client (page 9, lines 24-29 of the specification); sending the initial TCP state to the second BTCP module, including the initial sequence number, that enables the second BTCP module to understand a proper TCP state of the first BTCP module for the communication session (page 10,

lines 1-10 of the specification); receiving a SYN/ACK packet at the first BTCP module from the second TCP module, the SYN/ACK packet updated by the second BTCP module to reflect the proper TCP state at the first BTCP for the communication session (page 10, lines 11-13 of the specification); and sending the ACK packet to the selected back-end web server to extend the communication session to the selected server (page 10, lines 16-21 of the specification).

In one embodiment, such as that of dependent claim 14, the method further comprises sending response packets from the back-end web server to the client in a communication path that does not include the front-end node, by changing headers of the response packets such that it appears that the source of the response packets is the front-end node with the proper TCP state (*see* Figure 2, and *see* page 17, lines 5-15 of the specification).

In one embodiment, such as that of dependent claim 15, terminating the communication session further comprises intercepting TCP control packets from the selected back-end web server at the second BTCP module (*see* page 33, lines 8-11 of the specification); sending the TCP control packets to the first BTCP module from the second BTCP module (page 33, lines 8-11 of the specification); sending the TCP control packets to the client from the first BTCP module (page 33, lines 11-14 of the specification); and terminating the communication session at the front-end node and the back-end web server (*see* page 9, lines 4-15 of the specification).

In one embodiment, such as that of dependent claim 17, the method bypasses the first TCP module (*see* Figure 5, where TCP module 320 is bypassed).

According to another claimed embodiment, such as that of independent claim 22, a communication network for TCP state migration comprises a client (client 210 of Figure 2) and a front-end node (front-end node 252 of Figure 2) coupled to the client by the communication network. The front-end node includes a front-end bottom TCP (BTCP) module (BTCP module 350 of Figures 3C and 5, and *see* page 7, lines 16-29 and page 24, lines 13-25 of the specification) located below a front-end TCP module (TCP module 320 of Figures 3B, 3C, and 5, and *see* page 7, lines 16-24 and page 19, lines 28 through page 20, line 27 of the specification) in a first operating system, and a bottom IP (BIP) module (BIP module 360 of Figures 3C and 5,

and *see* page 24, lines 13-25 of the specification) located below an IP module (IP module 330 of Figures 3B, 3C, and 5, and see page 19, line 28 through page 20, lines 27 of the specification) in the first operating system. The communication network further comprises a plurality of back-end web servers (back-end web servers 255, 257, and 259 of Figure 2, and *see* page 7, lines 5-15 of the specification) including a selected back-end web server. The plurality of back-end web servers contain content that is partitioned between each of the plurality of back-end web servers. Each of the plurality of back-end web servers is coupled to the front-end node through the communication network. Each of the plurality of back-end web servers includes a back-end bottom TCP module (BTCP module 520 of Figure 5) located below a back-end TCP module (TCP module 530 of Figure 5).

In one embodiment, such as that of dependent claim 23, the front-end BTCP module establishes a communication session with the client for the transfer of data contained within the content to the client (page 7, lines 16-29 of the specification).

In one embodiment, such as that of dependent claim 24, the front-end BTCP module parses a HTTP request from the client in order to determine which of the plurality of back-end web servers, a selected back-end web server, contains the data in order to process the HTTP request (page 8, lines 1-10 of the specification).

In one embodiment, such as that of dependent claim 25, the front-end BTCP module extends the communication session to the selected back-end web server by handing-off an initial TCP state of the front-end BTCP module to a second BTCP module (BTCP module 520 of Figure 5) located at the selected back-end web server (page 8, lines 11-15 of the specification). The initial TCP state is associated with a proper TCP state for the front-end BTCP module in the communication session. The front-end BTCP module further forwards packets, including the HTTP request, from the client after successfully handing-off the initial TCP state (page 8, line 16 – page 9, line 2 of the specification).

In one embodiment, such as that of dependent claim 26, the second BTCP module understands the proper TCP state of the front-end BTCP module in the communication session

and modifies headers in response packets from the selected back-end web server to reflect the proper TCP state (page 10, lines 1-10 of the specification).

In one embodiment, such as that of dependent claim 27, the BIP module changes a destination address in forwarding the packets from the client (page 32, lines 10 – page 33, line 6 of the specification).

In one embodiment, such as that of dependent claim 28, the second BTCP module located at the selected back-end web server sends the response packets from the selected back-end web server to the client in a communication path that does not include the front-end node by changing headers of the response packets such that it appears the source of the response packets is the front-end node (*see* Figure 2, and *see* page 17, lines 5-15 of the specification).

VI. GROUNDS OF REJECTION TO BE REVIEWED ON APPEAL

Claims 1-29 are rejected under 35 U.S.C. § 103(a) as being unpatentable over U.S. Patent No. 6,775,692 issued to Albert et al (hereinafter "*Albert*") in view of U.S. Patent No. 5,774,660 issued to Brendel et al. (hereinafter "*Brendel*").

VII. ARGUMENT

Appellant respectfully traverses the outstanding rejections of the pending claims, and requests that the Board reverse the outstanding rejections in light of the remarks contained herein. Below, Appellant argues many of the rejected claims separately. Thus, Appellant respectfully asserts that separately argued claims do not stand or fall together, *see* 37 C.F.R. § 41.37(c)(1)(vii).

All of claims 1-29 were previously rejected in a Final Office Action mailed April 20, 2005 as being anticipated under 35 U.S.C. § 102(e) by *Albert*. In response, Applicant appealed the rejection to the Board and submitted an Appeal Brief presenting arguments regarding why the claims are not anticipated by *Albert*. In response to the Appeal Brief, the Examiner has reopened prosecution and now rejects the claims as being unpatentable over *Albert* in view of *Brendel*.

Appellant respectfully submits that *Brendel* does not cure the deficiencies of *Albert* for the reasons discussed below. In particular, *Brendel* is discussed in the Background section of the present application (*see* page 4, line 10 – page 5, line 15 of the present application) and is noted as disclosing an inefficient mechanism for transferring TCP states that requires use of a proprietary protocol that is known only to the application level, which embodiments of the present invention overcome. Thus, for the reasons discussed further below, the combination of *Brendel* with *Albert* fails to render the claims unpatentable. As such, Appellant respectfully requests that the rejections be overturned and this application be passed to allowance.

The test for non-obvious subject matter is whether the differences between the subject matter and the prior art are such that the claimed subject matter as a whole would have been obvious to a person having ordinary skill in the art to which the subject matter pertains. The United States Supreme Court in Graham v. John Deere and Co., 383 U.S. 1 (1966) set forth the factual inquiries which must be considered in applying the statutory test: (1) determining of the scope and content of the prior art; (2) ascertaining the differences between the prior art and the claims at issue; and (3) resolving the level of ordinary skill in the pertinent art. As discussed further hereafter, Appellant respectfully asserts that the claims include non-obvious differences over the cited art.

As discussed further below, when considering the scope and content of the applied *Albert* and *Brendel* references there are significant differences between the applied combination and the claims, as the applied combination fails to disclose all elements of the claims. Thus, considering the lack of disclosure in the applied combination of all elements of the claims, one of ordinary skill in the art would not find the claims obvious under 35 U.S.C. §103, and therefore the rejections should be overturned.

Independent Claim 1

Independent claim 1 recites:

> In a communication network, a method of TCP state migration comprising the steps of:

a) establishing a communication session between a client and a front-end node at a first bottom TCP (BTCP) module located below a first TCP module in a first operating system at said front-end node, said front-end node accessing a plurality of back-end web servers forming a web server cluster that contains content;

b) receiving a HTTP request from said client at said first BTCP module;

c) parsing said HTTP request to determine which back-end web server, a selected back-end web server, in said plurality of back-end web servers can process said HTTP request, said selected back-end web server not said front-end node;

d) extending said communication session to said selected back-end web server by handing-off an initial TCP state of said first BTCP module to said selected back-end web server;

e) sending said HTTP request to said selected back-end web server;

f) switching a bottom IP (BIP) module at said front-end node to a forwarding mode, wherein packets received at said BIP module from said client are forwarded to said selected back-end web server, said BIP module located below an IP module at said front-end node; and

g) terminating said communication session at said front-end node after said HTTP request is fully processed. (Emphasis added).

The combination of *Albert* and *Brendel* fails to teach or suggest all elements of independent claim 1. For the reasons discussed further in the Appeal Brief of January 5, 2006, *Albert* fails to disclose at least:

A) a first bottom TCP (BTCP) module located below a first TCP module in a first operating system at a front-end node;

B) parsing an HTTP request to determine which back-end web server can process the HTTP request;

C) a bottom IP (BIP) module located below an IP module at the front-end node; and

D) handing-off an initial TCP state of said first BTCP module to said selected back-end web server.

The Final Office Action appears to concede that *Albert* fails to teach or suggest these elements, *see* page 3 of the Final Office Action. However, the Final Office Action asserts that *Brendel* discloses these elements of the claim. Appellant respectfully disagrees, as discussed below.

The present application briefly discusses *Brendel* at page 4, line 10 – page 5, line 15 as follows:

> Another solution for request distribution is illustrated by the Brendel et al patent (U.S. 5,774,660) by Resonate, Inc. In Brendel et al., a load balancer examines the content of the web request to provide for better efficiency in processing requests. However, the Brendel et al. patent platform weaves a proprietary protocol within the TCP/IP protocol of an operating system of the load balancer. As a result, the algorithm utilized by the Brendel et al. patent necessitate kernel source modifications when porting from one operating system to another.
>
> Also, in the Brendel et al. patent the proprietary protocol is applied at the application layer of the operating system of the load balancer. Incoming packets to the load balancer have their protocol changed from TCP to a non-TCP (IXP) standard that is only understood by the proprietary protocol located at the application layer. Later, the packets have their packets changed back to the TCP protocol for transmission to the back-end servers. Thus, the Brendel et al. patent reduces processing efficiency by switching back and forth between user level kernels.. Further, were the Brendel et al. patent to be implemented at the operating system's kernel level, any modifications made to the proprietary protocol would necessarily require access to the kernel source file which typically is not available.
>
> Thus, a need exists for more flexibility in designing and implementing a TCP/IP handoff mechanism in a web server cluster. Another need exists for a TCP/IP handoff mechanism that is more portable between different operating systems implementing the TCP/IP protocol. Still another need exists for better efficiency in performing TCP/IP handoff mechanisms.

Thus, the present application expressly recognized that *Brendel* fails to provide modules within an operating system, such as those recited above in claim 1. For instance, *Brendel* does not teach or suggest "handing-off an initial TCP state of said first BTCP module to said selected back-end web server". No such BTCP module is implemented in the operating system of *Brendel*. Instead, *Brendel* requires that incoming packets be changed into a proprietary protocol that is understood only at the application layer. For instance, *Brendel* explains at col. 13, lines 40-46 thereof:

> Modified TCP/IP stack 82 contains the standard TCP and IP modules with some modifications explained later. One modification is that incoming packets

from the Internet have their protocol changed from TCP to a proprietary "IXP" protocol. Since this IXP protocol is unknown to the standard TCP and IP layers, it is sent directly up to application layer 80 containing the load balancer.

Thus, *Brendel* appears to disclose a system in which the TCP/IP stack of an operating system is modified so as to change incoming packets from the TCP protocol to a proprietary protocol that is understood only at the application layer, rather than implementing modules, such as a BTCP module within the operating system, as recited by claim 1.

Thus, for at least the above reasons, the combination of *Albert* and *Brendel* fails to teach or suggest all elements of claim 1. As such, the rejection of claim 1 should be overturned, and claim 1 should be passed to allowance.

Independent Claim 11

Independent claim 11 recites:

In a communication network, a method of TCP state migration comprising the steps of:

a) receiving a request from a client for establishing a communication session at a first bottom TCP (BTCP) module located at a front-end node, said front-end node accessing a plurality of back-end web servers containing content, wherein said content is partially replicated between each of said plurality of back-end web servers, said communication session established for the transfer of data contained within said content to said client;

b) establishing said communication session between said client and said first BTCP module, said first BTCP module located below a first TCP module in a first operating system at said front-end node;

c) receiving a HTTP request from said client at said first BTCP module;

d) parsing said HTTP request to determine which back-end web server, a selected back-end web server, in said plurality of back-end web servers contains said data in order to process said HTTP request, said selected back-end web server not said front-end node;

e) extending said communication session to said selected back-end web server by handing-off an initial TCP state of said first BTCP module to a second BTCP module located at said selected back-end web server, said initial TCP state associated with said communication session between said client and said first BTCP module, said second BTCP module located below a second TCP

module in a second operating system at said selected back-end web server;

f) sending said HTTP request to said selected back-end web server;

g) switching a bottom IP (BIP) module in said front-end node to a forwarding mode, wherein packets, from said client, received at said front-end node are intercepted by said BIP module and forwarded to said selected back-end web server, said BIP module located below an IP module in said front-end node, said BIP module changing destination IP addresses of said packets to said selected back-end web server and

h) terminating said communication session after said HTTP request has been fully processed.

The combination of *Albert* and *Brendel* fails to teach or suggest all elements of independent claim 11. For the reasons discussed further in the Appeal Brief of January 5, 2006, *Albert* fails to disclose at least:

A) a first BTCP module located below a first TCP module in a first operating system at a front-end node;

B) parsing an HTTP request to determine which back-end web server contains data in order to process the HTTP request;

C) a bottom IP (BIP) module located below an IP module in the front-end node;

D) handing-off an initial TCP state of said first BTCP module to a second BTCP module located at said selected back-end web server; and

E) a second BTCP module located below a second TCP module in a second operating system at said selected back-end web server.

Further, *Brendel* fails to teach or suggest at least first and second BTCP modules, and handing-off an initial TCP state of the first BTCP module to the second BTCP module, as recited by claim 11. For instance, as discussed above with claim 1, *Brendel* does not teach or suggest any such BTCP modules, but instead appears to disclose a modified TCP/IP stack that changes an incoming packet's protocol to a proprietary protocol that is unknown to the TCP/IP stack for handling at the application level.

Thus, for at least the above reasons, the combination of *Albert* and *Brendel* fails to teach or suggest all elements of claim 11. As such, the rejection of claim 11 should be overturned, and claim 11 should be passed to allowance.

Independent Claim 22

Independent claim 22 recites:

> A communication network for TCP state migration comprising:
> a client;
> a front-end node coupled to said client by said communication network, said front-end node including a front-end bottom TCP (BTCP) module located below a front-end TCP module in a first operating system, and a bottom IP (BIP) module located below an IP module in said first operating system; and
> a plurality of back-end web servers including a selected back-end web server, said plurality of back-end web servers containing content that is partitioned between each of said plurality of back-end web servers, each of said plurality of back-end web servers coupled to said front-end node through said communication network, each of said plurality of back-end web servers including a back-end bottom TCP module located below a back-end TCP module.

The combination of *Albert* and *Brendel* fails to teach or suggest all elements of independent claim 22. For the reasons discussed further in the Appeal Brief of January 5, 2006, *Albert* fails to disclose at least:

A) a front-end node that includes a front-end bottom TCP (BTCP) module located below a front-end TCP module in a first operating system;

B) a front-end node that further includes a bottom IP (BIP) module located below an IP module in said first operating system; and

C) a back-end web server that includes a back-end bottom TCP module located below a back-end TCP module.

Further, *Brendel* fails to teach or suggest at least the BTCP and BIP modules, as recited by claim 22. For instance, as discussed above with claim 1, *Brendel* does not teach or suggest any such BTCP modules, but instead appears to disclose a modified TCP/IP stack that changes an incoming packet's protocol to a proprietary protocol that is unknown to the TCP/IP stack for handling at the application level.

Thus, for at least the above reasons, the combination of *Albert* and *Brendel* fails to teach or suggest all elements of claim 22. As such, the rejection of claim 22 should be overturned, and claim 22 should be passed to allowance.

Dependent Claims 2, 4, and 8-10

Claims 2, 4, and 8-10 each depend either directly or indirectly from independent claim 1, and thus claims 2, 4, and 8-10 each inherit all elements of claim 1. Therefore, claims 2, 4, and 8-10 are each allowable over the applied combination of *Albert* and *Brendel* at least for the reasons discussed above with claim 1. As such, Appellant respectfully requests that the rejection of claims 2, 4 and 8-10 be overturned.

Dependent Claim 3

Dependent claim 3 depends from claim 1 and thus inherits all elements of claim 1. Accordingly, claim 3 is allowable over *Albert* in view of *Brendel* at least for the reasons discussed above with claim 1. Additionally, claim 3 further recites "wherein said back-end web server includes a second BTCP module that is located below a second TCP module in a second operating system" (emphasis added).

The combination of *Albert* in view of *Brendel* fails to disclose this further element of claim 3. The Final Office Action appears to rely upon *Albert* as disclosing this further element of claim 3 (*see* page 4 of the Final Office Action), as the reasoning for rejecting claim 3 appears to be the same as in the Final Office Action of April 20, 2005 (in which claim 3 was rejected as being anticipated by *Albert*, *see* page 4 of the Final Office Action of April 20, 2005). However, *Albert* fails to disclose this further element of claim 3. As discussed above with claims 11 and 22, *Albert* makes no mention of a second BTCP module that is located below a second TCP module in an operating system of a selected back-end server.

In view of the above, Appellant respectfully requests that the rejection of claim 3 be overturned.

Dependent Claim 5

Dependent claim 5 depends from claim 4, which depends from claim 1, and thus claim 5 inherits all elements of claims 1 and 4. Accordingly, claim 5 is allowable over the applied

combination of *Albert* and *Brendel* at least for the reasons discussed above with claims 1 and 4. Additionally, claim 5 further recites:

> The method as described in Claim 4, wherein said step d) comprises the further steps of:
> sending a SYN packet to said selected back-end web server, said SYN packet intercepted by a second BTCP module, said SYN packet originally sent from said client to said front-end node in requesting said communication session, said SYN packet stored at said first BTCP module;
> including an initial sequence number within said SYN packet that enables said second BTCP module to understand a proper TCP state of said first BTCP module in said communication session;
> receiving a SYN/ACK packet from said selected back-end web server, said SYN/ACK packet updated by said second BTCP module to reflect said proper TCP state of said first BTCP module; and
> sending an ACK packet from said first BTCP module to said selected back-end web server, said ACK packet originally sent from said client to said front-end node in establishing said communication session.

The combination of *Albert* in view of *Brendel* fails to disclose these further elements of claim 5. The Final Office Action appears to rely upon *Albert* as disclosing the further elements of claim 5 (*see* page 5 of the Final Office Action), as the reasoning for rejecting claim 5 appears to be the same as in the Final Office Action of April 20, 2005 (in which claim 5 was rejected as being anticipated by *Albert*, *see* pages 4-5 of the Final Office Action of April 20, 2005). However, *Albert* fails to disclose this further element of claim 5. As discussed above, *Albert* does not teach the recited first and second BTCP modules. Thus, for at least this reason, *Albert* further fails to teach the steps recited in claim 5. Accordingly, Appellant respectfully requests that the rejection of claim 5 be overturned.

Dependent Claim 6

Dependent claim 6 depends from claim 1 and thus inherits all elements of claim 1. Accordingly, claim 6 is allowable over the applied combination of *Albert* and *Brendel* at least for the reasons discussed above with claim 1. Additionally, claim 6 further recites:

> The method as described in Claim 1, wherein said method comprises the further step of:

> sending response packets from said selected back-end web server to said client in a communication path that does not include said front-end node by changing headers of said response packets such that it appears that the source of said response packets is said first BTCP in its proper TCP state.

The combination of *Albert* in view of *Brendel* fails to disclose these further elements of claim 6. The Final Office Action appears to rely upon *Albert* as disclosing the further elements of claim 6 (*see* pages 5-6 of the Final Office Action), as the reasoning for rejecting claim 6 appears to be the same as in the Final Office Action of April 20, 2005 (in which claim 6 was rejected as being anticipated by *Albert*, *see* page 5 of the Final Office Action of April 20, 2005). However, *Albert* fails to disclose this further element of claim 6. That is, *Albert* fails to teach that its back-end servers send a response to a client in a communication path that does not include the front-end node. The Final Office Action asserts that the forwarding agent of *Albert* is the recited front-end node. *Albert* does not teach that its back-end server responds to a client via a communication path that does not include the forwarding agent. Rather, the responsive communication from the back-end servers in *Albert* is sent back through the forwarding agents.

Accordingly, Appellant respectfully requests that the rejection of claim 6 be overturned.

Dependent Claim 7

Dependent claim 7 depends from claim 1 and thus inherits all elements of claim 1. Accordingly, claim 7 is allowable over the applied combination of *Albert* and *Brendel* at least for the reasons discussed above with claim 1. Additionally, claim 7 further recites:

> The method as described in Claim 1, wherein step g) comprises the further steps of:
> intercepting TCP control packets from a second TCP module located at said selected back-end web server at said second BTCP module;
> sending said TCP control packets to said first BTCP module from said second BTCP module;
> sending said TCP control packets to said client from said first BTCP module; and
> terminating said communication session at said front-end node and said back-end web server.

The combination of *Albert* in view of *Brendel* fails to disclose these further elements of claim 7. The Final Office Action appears to rely upon *Albert* as disclosing the further elements of claim 7 (*see* page 6 of the Final Office Action), as the reasoning for rejecting claim 7 appears to be the same as in the Final Office Action of April 20, 2005 (in which claim 7 was rejected as being anticipated by *Albert*, *see* pages 5-6 of the Final Office Action of April 20, 2005). However, *Albert* fails to disclose this further element of claim 7. As discussed above, *Albert* does not teach the recited first and second BTCP modules. Thus, for at least this reason, *Albert* further fails to teach the steps recited in claim 7. Accordingly, Appellant respectfully requests that the rejection of claim 7 be overturned.

Dependent Claims 13, 16, and 18-21

Claims 13, 16, and 18-21 each depend either directly or indirectly from independent claim 11, and thus claims 13, 16, and 18-21 each inherit all elements of claim 11. Therefore, claims 13, 16, and 18-21 are each allowable over the applied combination of *Albert* and *Brendel* at least for the reasons discussed above with claim 11. As such, Appellant respectfully requests that the rejection of claims 13, 16, and 18-21 be overturned.

Dependent Claim 12

Dependent claim 12 depends from claim 11 and thus inherits all elements of claim 11. Accordingly, claim 12 is allowable over the applied combination of *Albert* and *Brendel* at least for the reasons discussed above with claim 11. Additionally, claim 12 further recites:

> The method as described in Claim 11, wherein step e) comprises the further steps of:
> e1) storing a SYN packet sent from said client to said front-end node, said SYN packet requesting said communication session in step a);
> e2) storing an ACK packet sent from said client to said front end node in establishing said communication session;
> e3) sending said SYN packet to said selected back-end web server so that it appears that said SYN packet originated from said client;
> e4) sending said initial TCP state to said second BTCP module, including said initial sequence number, that enables said second BTCP module to understand a proper TCP state of said first BTCP module for said communication

session;

e5) receiving a SYN/ACK packet at said first BTCP module from said second TCP module, said SYN/ACK packet updated by said second BTCP module to reflect said proper TCP state at said first BTCP for said communication session; and

e6) sending said ACK packet to said selected back-end web server to extend said communication session to said selected server.

The combination of *Albert* in view of *Brendel* fails to disclose these further elements of claim 12. The Final Office Action appears to rely upon *Albert* as disclosing the further elements of claim 12 (*see* page 7 of the Final Office Action), as the reasoning for rejecting claim 12 appears to be the same as in the Final Office Action of April 20, 2005 (in which claim 12 was rejected as being anticipated by *Albert*). However, *Albert* fails to disclose the further elements of claim 12. As discussed above, *Albert* does not teach the recited first and second BTCP modules. Thus, for at least this reason, *Albert* further fails to teach the steps recited in claim 12 that involve those modules. Accordingly, Appellant respectfully requests that the rejection of claim 12 be overturned.

Dependent Claim 14

Dependent claim 14 depends from claim 11 and thus inherits all elements of claim 11. Accordingly, claim 14 is allowable over the applied combination of *Albert* and *Brendel* at least for the reasons discussed above with claim 11. Additionally, claim 14 further recites:

> The method as described in Claim 11, wherein said method comprises the further step of sending response packets from said back-end web server to said client in a communication path that does not include said front-end node, by changing headers of said response packets such that it appears that the source of said response packets is said front-end node with said proper TCP state.

The combination of *Albert* in view of *Brendel* fails to disclose these further elements of claim 14. The Final Office Action appears to rely upon *Albert* as disclosing the further elements of claim 14 (*see* page 7 of the Final Office Action), as the reasoning for rejecting claim 14 appears to be the same as in the Final Office Action of April 20, 2005 (in which claim 14 was rejected as being anticipated by *Albert*). However, *Albert* fails to disclose the further elements of claim 14. That is, *Albert* fails to teach that its back-end servers send a response to a client in a

communication path that does not include the front-end node. The Final Office Action asserts that the forwarding agent of *Albert* is the recited front-end node. *Albert* does not teach that its back-end server responds to a client via a communication path that does not include the forwarding agent. Rather, the responsive communication from the back-end servers in *Albert* is sent back through the forwarding agents.

Accordingly, Appellant respectfully requests that the rejection of claim 14 be overturned.

Dependent Claim 15

Dependent claim 15 depends from claim 11 and thus inherits all elements of claim 11. Accordingly, claim 15 is allowable over the combination of *Albert* and *Brendel* at least for the reasons discussed above with claim 11. Additionally, claim 15 further recites:

> The method as described in Claim 11, wherein step h) comprises the steps of:
> intercepting TCP control packets from said selected back-end web server at said second BTCP module;
> sending said TCP control packets to said first BTCP module from said second BTCP module;
> sending said TCP control packets to said client from said first BTCP module; and
> terminating said communication session at said front-end node and said back-end web server.

The combination of *Albert* in view of *Brendel* fails to disclose these further elements of claim 15. The Final Office Action appears to rely upon *Albert* as disclosing the further elements of claim 15 (*see* page 7 of the Final Office Action), as the reasoning for rejecting claim 15 appears to be the same as in the Final Office Action of April 20, 2005 (in which claim 15 was rejected as being anticipated by *Albert*). However, *Albert* fails to disclose the further elements of claim 15. As discussed above, *Albert* does not teach the recited first and second BTCP modules. Thus, for at least this reason, *Albert* further fails to teach the steps recited in claim 15 that involve those modules. Accordingly, Appellant respectfully requests that the rejection of claim 15 be overturned.

Dependent Claim 17

Dependent claim 17 depends from claim 11 and thus inherits all elements of claim 11. Accordingly, claim 17 is allowable over the applied combination of *Albert* and *Brendel* at least for the reasons discussed above with claim 11. Additionally, claim 17 further recites:

> The method as described in Claim 11, wherein said method bypasses the first TCP module.

The combination of *Albert* in view of *Brendel* fails to disclose this further element of claim 17. The Final Office Action appears to rely upon *Albert* as disclosing the further elements of claim 17 (*see* page 7 of the Final Office Action), as the reasoning for rejecting claim 17 appears to be the same as in the Final Office Action of April 20, 2005 (in which claim 17 was rejected as being anticipated by *Albert*). However, *Albert* fails to disclose the further elements of claim 17. For instance, *Albert* fails to teach a method that bypasses a TCP module at the front-end node (e.g., forwarding agent of *Albert*). Thus, for at least this reason, Appellant respectfully requests that the rejection of claim 17 be overturned.

Dependent Claim 29

Claim 29 depends from independent claim 22 and thus inherits all elements of claim 22. Therefore, claim 29 is allowable over the applied combination of *Albert* and *Brendel* at least for the reasons discussed above with claim 22. As such, Appellant respectfully requests that the rejection of claim 29 be overturned.

Dependent Claim 23

Dependent claim 23 depends from claim 22 and thus inherits all elements of claim 22. Accordingly, claim 23 is allowable over the applied combination of *Albert* and *Brendel* at least for the reasons discussed above with claim 22. Additionally, claim 23 further recites:

> The communication network as described in Claim 22, wherein said front-end BTCP module establishes a communication session with said client for the transfer of data contained within said content to said client.

The combination of *Albert* in view of *Brendel* fails to disclose this further element of claim 23. The Final Office Action appears to rely upon *Albert* as disclosing the further elements of claim 23 (*see* page 7 of the Final Office Action), as the reasoning for rejecting claim 23 appears to be the same as in the Final Office Action of April 20, 2005 (in which claim 23 was rejected as being anticipated by *Albert*). However, *Albert* fails to disclose the further elements of claim 23. As discussed above, *Albert* does not teach the recited front-end BTCP module, and thus fails to teach establishing a communication session with a client as recited by this element of claim 23. Accordingly, Appellant respectfully requests that the rejection of claim 23 be overturned.

Dependent Claim 24

Dependent claim 24 depends from claim 23, which depends from claim 22, and thus claim 24 inherits all elements of claims 22 and 23. Accordingly, claim 24 is allowable over the applied combination of *Albert* and *Brendel* at least for the reasons discussed above with claims 22-23. Additionally, claim 24 further recites:

> The communication network as described in Claim 23, wherein said front-end BTCP module parses a HTTP request from said client in order to determine which of said plurality of back-end web servers, a selected back-end web server, contains said data in order to process said HTTP request.

The combination of *Albert* in view of *Brendel* fails to disclose this further element of claim 24. The Final Office Action appears to rely upon *Albert* as disclosing the further element of claim 24 (*see* page 7 of the Final Office Action), as the reasoning for rejecting claim 24 appears to be the same as in the Final Office Action of April 20, 2005 (in which claim 24 was rejected as being anticipated by *Albert*). However, *Albert* fails to disclose the further elements of claim 24. For instance, as discussed above with claims 1 and 11, *Albert* does not teach parsing a HTTP request from a client in order to determine which of the plurality of back-end web servers contains the data. Accordingly, Appellant respectfully requests that the rejection of claim 24 be overturned.

Dependent Claim 25

Dependent claim 25 depends from claim 23, which depends from claim 22, and thus claim 25 inherits all elements of claims 22 and 23. Accordingly, claim 25 is allowable over the applied combination of *Albert* and *Brendel* at least for the reasons discussed above with claims 22-23. Additionally, claim 25 further recites:

> The communication network as described in Claim 23, wherein said front-end BTCP module extends said communication session to said selected back-end web server by handing-off an initial TCP state of said front-end BTCP module to a second BTCP module located at said selected back-end web server, said initial TCP state associated with a proper TCP state for said front-end BTCP module in said communication session, said front-end BTCP module further forwarding packets, including said HTTP request, from said client after successfully handing-off said initial TCP state.

The combination of *Albert* in view of *Brendel* fails to disclose these further elements of claim 25. The Final Office Action appears to rely upon *Albert* as disclosing the further elements of claim 25 (*see* page 7 of the Final Office Action), as the reasoning for rejecting claim 25 appears to be the same as in the Final Office Action of April 20, 2005 (in which claim 25 was rejected as being anticipated by *Albert*). However, *Albert* fails to disclose the further elements of claim 25. As discussed above, *Albert* does not teach the recited front-end BTCP module. *Albert* further fails to teach the recited second BTCP module located at the selected back-end web server. Thus, Appellant respectfully requests that the rejection of claim 25 be overturned.

Dependent Claim 26

Dependent claim 26 depends from claim 25, which depends from claim 23 which depends from claim 22, and thus claim 26 inherits all elements of claims 22-23 and 25. Accordingly, claim 26 is allowable over the applied combination of *Albert* and *Brendel* at least for the reasons discussed above with claims 22-23 and 25. Additionally, claim 26 further recites:

> The communication network as described in Claim 25, wherein said second BTCP module understands said proper TCP state of said front-end BTCP module in said communication session and modifies headers in response packets from said selected back-end web server to reflect said proper TCP state.

The combination of *Albert* in view of *Brendel* fails to disclose these further elements of claim 26. The Final Office Action appears to rely upon *Albert* as disclosing the further elements of claim 26 (*see* page 7 of the Final Office Action), as the reasoning for rejecting claim 26 appears to be the same as in the Final Office Action of April 20, 2005 (in which claim 26 was rejected as being anticipated by *Albert*). However, *Albert* fails to disclose the further elements of claim 26. For instance, as discussed above, *Albert* does not teach the recited second BTCP module or the front-end BTCP module. *Albert* further fails to teach such a second BTCP that understands the proper TCP state of the front-end BTCP module and modifies headers in response packets as recited in claim 26. Thus, Appellant respectfully requests that the rejection of claim 26 be overturned.

Dependent Claim 27

Dependent claim 27 depends from claim 25, which depends from claim 23 which depends from claim 22, and thus claim 27 inherits all elements of claims 22-23 and 25. Accordingly, claim 27 is allowable over the applicd combination of *Albert* in view of *Brendel* at least for the reasons discussed above with claims 22-23 and 25. Additionally, claim 27 further recites:

> The communication network as described in Claim 25, wherein said BIP module changes a destination address in forwarding said packets from said client.

The combination of *Albert* in view of *Brendel* fails to disclose these further elements of claim 27. The Final Office Action appears to rely upon *Albert* as disclosing the further elements of claim 27 (*see* page 7 of the Final Office Action), as the reasoning for rejecting claim 27 appears to be the same as in the Final Office Action of April 20, 2005 (in which claim 27 was rejected as being anticipated by *Albert*). However, *Albert* fails to disclose the further elements of claim 27. As discussed above, *Albert* does not teach the recited BIP module. *Albert* further fails to teach such a BIP module that changes a destination address in forwarding packets from the client, as recited in claim 27. Thus, Appellant respectfully requests that the rejection of claim 27 be overturned.

Dependent Claim 28

Dependent claim 28 depends from claim 26, which depends from claim 25 which depends from claim 23 which depends from claim 22, and thus claim 28 inherits all elements of claims 22-23 and 25-26. Accordingly, claim 28 is allowable over the applied combination of *Albert* and *Brendel* at least for the reasons discussed above with claims 22-23 and 25-26. Additionally, claim 28 further recites:

> The communication network, as described in Claim 26, wherein said second BTCP module located at said selected back-end web server sends said response packets from said selected back-end web server to said client in a communication path that does not include said front-end node by changing headers of said response packets such that it appears the source of said response packets is said front-end node.

The combination of *Albert* in view of *Brendel* fails to disclose these further elements of claim 28. The Final Office Action appears to rely upon *Albert* as disclosing the further elements of claim 28 (*see* page 7 of the Final Office Action), as the reasoning for rejecting claim 28 appears to be the same as in the Final Office Action of April 20, 2005 (in which claim 28 was rejected as being anticipated by *Albert*). However, *Albert* fails to disclose the further elements of claim 28. That is, *Albert* fails to teach that its back-end servers send a response to a client in a communication path that does not include the front-end node. The Final Office Action asserts that the forwarding agent of *Albert* is the recited front-end node. *Albert* does not teach that its back-end server responds to a client via a communication path that does not include the forwarding agent. Rather, the responsive communication from the back-end servers in *Albert* is sent back through the forwarding agents.

Accordingly, Appellant respectfully requests that the rejection of claim 28 be overturned.

Conclusion

In view of the above, Appellant requests that the board overturn the outstanding rejections of claims 1-29. Attached hereto are a Claims Appendix, Evidence Appendix, and Related Proceedings Appendix. As noted in the attached Evidence Appendix, no evidence pursuant to §§ 1.130, 1.131, or 1.132 or entered by or relied upon by the examiner is being submitted. Also, certain related appeals are identified in Section II above, but as noted by the Related Proceedings Appendix, no decisions have been received in such appeals and thus no copies of any decisions in related proceedings are provided.

I hereby certify that this paper (along with any paper referred to as being attached or enclosed) is being transmitted via the Office electronic filing system in accordance with § 1.6(a)(4).

Dated: July 2, 2007

Signature: Donna Forbit
(Donna Forbit)

Respectfully submitted,

By: _____
Jody C. Bishop
Attorney/Agent for Applicant(s)
Reg. No. 44,034
Date: July 2, 2007
Telephone No. (214) 855-8007

VIII. CLAIMS APPENDIX

Claims Involved in the Appeal of Application Serial No. 09/880,632

1. In a communication network, a method of TCP state migration comprising the steps of:

a) establishing a communication session between a client and a front-end node at a first bottom TCP (BTCP) module located below a first TCP module in a first operating system at said front-end node, said front-end node accessing a plurality of back-end web servers forming a web server cluster that contains content;

b) receiving a HTTP request from said client at said first BTCP module;

c) parsing said HTTP request to determine which back-end web server, a selected back-end web server, in said plurality of back-end web servers can process said HTTP request, said selected back-end web server not said front-end node;

d) extending said communication session to said selected back-end web server by handing-off an initial TCP state of said first BTCP module to said selected back-end web server;

e) sending said HTTP request to said selected back-end web server;

f) switching a bottom IP (BIP) module at said front-end node to a forwarding mode, wherein packets received at said BIP module from said client are forwarded to said selected back-end web server, said BIP module located below an IP module at said front-end node; and

g) terminating said communication session at said front-end node after said HTTP request is fully processed.

2. The method as described in Claim 1, wherein said content is partially replicated between each of said plurality of back-end web servers.

3. The method as described in Claim 1, wherein said back-end web server includes a second BTCP module that is located below a second TCP module in a second operating system.

4. The method as described in Claim 1, wherein said initial TCP state is associated with said communication session, said communication session established for the transfer of data contained within said content to said client.

5. The method as described in Claim 4, wherein said step d) comprises the further steps of:

sending a SYN packet to said selected back-end web server, said SYN packet intercepted by a second BTCP module, said SYN packet originally sent from said client to said front-end node in requesting said communication session, said SYN packet stored at said first BTCP module;

including an initial sequence number within said SYN packet that enables said second BTCP module to understand a proper TCP state of said first BTCP module in said communication session;

receiving a SYN/ACK packet from said selected back-end web server, said SYN/ACK packet updated by said second BTCP module to reflect said proper TCP state of said first BTCP module; and

sending an ACK packet from said first BTCP module to said selected back-end web server, said ACK packet originally sent from said client to said front-end node in establishing said communication session.

6. The method as described in Claim 1, wherein said method comprises the further step of:

sending response packets from said selected back-end web server to said client in a communication path that does not include said front-end node by changing headers of said response packets such that it appears that the source of said response packets is said first BTCP in its proper TCP state.

7. The method as described in Claim 1, wherein step g) comprises the further steps of:

intercepting TCP control packets from a second TCP module located at said selected back-end web server at said second BTCP module;

sending said TCP control packets to said first BTCP module from said second BTCP module;

sending said TCP control packets to said client from said first BTCP module; and

terminating said communication session at said front-end node and said back-end web server.

8. The method as described in Claim 1, wherein said front-end node and said plurality of back-end web servers comprise a web site, said front-end node providing a virtual IP address for said web site.

9. The method as described in claim 8, wherein said front-end node, and said plurality of back-end web servers are coupled together by a local area network.

10. The method as described in Claim 8, wherein said front-end node and said plurality of back-end web servers are coupled together by a wide area network,.

11. In a communication network, a method of TCP state migration comprising the steps of:

a) receiving a request from a client for establishing a communication session at a first bottom TCP (BTCP) module located at a front-end node, said front-end node accessing a plurality of back-end web servers containing content, wherein said content is partially replicated between each of said plurality of back-end web servers, said communication session established for the transfer of data contained within said content to said client;

b) establishing said communication session between said client and said first BTCP module, said first BTCP module located below a first TCP module in a first operating system at said front-end node;

c) receiving a HTTP request from said client at said first BTCP module;

d) parsing said HTTP request to determine which back-end web server, a selected back-end web server, in said plurality of back-end web servers contains said data in order to process said HTTP request, said selected back-end web server not said front-end node;

e) extending said communication session to said selected back-end web server by handing-off an initial TCP state of said first BTCP module to a second BTCP module located at said selected back-end web server, said initial TCP state associated with said communication session between said client and said first BTCP module, said second BTCP module located below a second TCP module in a second operating system at said selected back-end web server;

f) sending said HTTP request to said selected back-end web server;

g) switching a bottom IP (BIP) module in said front-end node to a forwarding mode, wherein packets, from said client, received at said front-end node are intercepted by said BIP module and forwarded to said selected back-end web server, said BIP module located below an IP module in said front-end node, said BIP module changing destination IP addresses of said packets to said selected back-end web server and

h) terminating said communication session after said HTTP request has been fully processed.

12. The method as described in Claim 11, wherein step e) comprises the further steps of:

e1) storing a SYN packet sent from said client to said front-end node, said SYN packet requesting said communication session in step a);

e2) storing an ACK packet sent from said client to said front end node in establishing said communication session;

e3) sending said SYN packet to said selected back-end web server so that it appears that said SYN packet originated from said client;

e4) sending said initial TCP state to said second BTCP module, including said initial sequence number, that enables said second BTCP module to understand a proper TCP state of said first BTCP module for said communication session;

e5) receiving a SYN/ACK packet at said first BTCP module from said second TCP module, said SYN/ACK packet updated by said second BTCP module to reflect said proper TCP state at said first BTCP for said communication session; and

e6) sending said ACK packet to said selected back-end web server to extend said communication session to said selected server.

13. The method as described in Claim 12, wherein step e4) includes the further step of including said initial sequence number in said SYN packet.

14. The method as described in Claim 11, wherein said method comprises the further step of sending response packets from said back-end web server to said client in a communication path that does not include said front-end node, by changing headers of said response packets such that it appears that the source of said response packets is said front-end node with said proper TCP state.

15. The method as described in Claim 11, wherein step h) comprises the steps of:

intercepting TCP control packets from said selected back-end web server at said second BTCP module;

sending said TCP control packets to said first BTCP module from said second BTCP module;

sending said TCP control packets to said client from said first BTCP module; and

terminating said communication session at said front-end node and said back-end web server.

16. The method as described in Claim 15, wherein said TCP control packets include a RST flag and a FIN flag.

17. The method as described in Claim 11, wherein said method bypasses the first TCP module.

18. The method as described in Claim 11, wherein said front-end node, and said plurality of back-end web servers comprise a web site, said front-end node providing a virtual IP address for said web site.

19. The method as described in claim 18, wherein said front-end node, and said plurality of back-end web servers are coupled together by a local area network.

20. The method as described in Claim 18, wherein said front-end node and said plurality of back-end web servers are coupled together by a wide area network.

21. The method as described in Claim 11, wherein said content is partitioned between each of said plurality of back-end web servers.

22. A communication network for TCP state migration comprising:

a client;

a front-end node coupled to said client by said communication network, said front-end node including a front-end bottom TCP (BTCP) module located below a front-end TCP module in a first operating system, and a bottom IP (BIP) module located below an IP module in said first operating system; and

a plurality of back-end web servers including a selected back-end web server, said plurality of back-end web servers containing content that is partitioned between each of said plurality of back-end web servers, each of said plurality of back-end web servers coupled to said front-end node through said communication network, each of said plurality of back-end web servers including a back-end bottom TCP module located below a back-end TCP module.

23. The communication network as described in Claim 22, wherein said front-end BTCP module establishes a communication session with said client for the transfer of data contained within said content to said client.

24. The communication network as described in Claim 23, wherein said front-end BTCP module parses a HTTP request from said client in order to determine which of said plurality of back-end web servers, a selected back-end web server, contains said data in order to process said HTTP request.

25. The communication network as described in Claim 23, wherein said front-end BTCP module extends said communication session to said selected back-end web server by handing-off an initial TCP state of said front-end BTCP module to a second BTCP module located at said selected back-end web server, said initial TCP state associated with a proper TCP state for said front-end BTCP module in said communication session, said front-end BTCP module further forwarding packets, including said HTTP request, from said client after successfully handing-off said initial TCP state.

26. The communication network as described in Claim 25, wherein said second BTCP module understands said proper TCP state of said front-end BTCP module in said communication session and modifies headers in response packets from said selected back-end web server to reflect said proper TCP state.

27. The communication network as described in Claim 25, wherein said BIP module changes a destination address in forwarding said packets from said client.

28. The communication network, as described in Claim 26, wherein said second BTCP module located at said selected back-end web server sends said response packets from said selected back-end web server to said client in a communication path that does not include said front-end node by changing headers of said response packets such that it appears the source of said response packets is said front-end node.

29. The communication network as described in Claim 22 wherein said content is partially replicated between each of said plurality of back-end web servers.

IX. EVIDENCE APPENDIX

No evidence pursuant to §§ 1.130, 1.131, or 1.132 or entered by or relied upon by the examiner is being submitted.

X. RELATED PROCEEDINGS APPENDIX

A currently pending appeal before the Board of co-pending U.S. Patent Application Serial No. 09/880,631, and particularly the Board's interpretation of *Albert* and *Brendel*, may affect, be affected by, or have a bearing on the Board's decision in this appeal.

As of the filing of this Appeal Brief, no decisions have been received in this related appeal, and thus no copies of such decisions in the related proceeding are provided.